Maven的多模块开发学习

<https://www.cnblogs.com/xdp-gacl/p/4242221.html>

# 引入：分层开发

　 在平时的Javaweb项目开发中为了便于后期的维护，我们一般会进行**分层开发**，最常见的就是分为**domain（域模型层）、dao（数据库访问层）、service（业务逻辑层）、web（表现层）**，这样分层之后，各个层之间的职责会比较明确，后期维护起来也相对比较容易，今天我们就是使用Maven来构建以上的各个层。

项目结构如下：

　　system-parent

　　　　|----pom.xml

　　　　|----system-domain

　　　　　　　　|----pom.xml

　　　　|----system-dao

　　　　　　　　|----pom.xml

　　　　|----system-service

　　　　　　　　|----pom.xml

　　　　|----system-web

　　　　　　　　|----pom.xml

# 创建system-parent项目

创建项目的时候，如果普通Java项目–DarchetypeArtifactId指定为maven-archetype-quickstart，若需要为web项目，指定为**maven-archetype-webapp**。这里利用web项目。

创建system-parent，**用来给各个子模块继承**。

　　进入命令行，输入以下命令：

mvn archetype: generate -DgroupId=me.gacl -DartifactId=system-parent –DarchetypeArtifactId = **maven-archetype-webapp** -DinteractiveMode=false

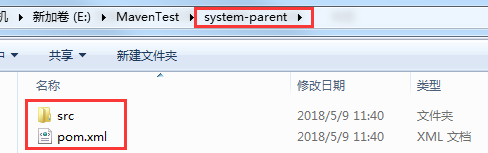
若失败：通过交互模式执行mvn archetype: generate；在交互模式中指定具体设置。

mvn archetype:generate -DarchetypeArtifactId=maven-archetype-webapp

将src文件夹删除，然后修改pom.xml文件，将<packaging>jar</packaging>修改为<packaging>pom</packaging>，pom表示它是一个被继承的模块，修改后的内容如下：

执行完毕之后，会在当前目录(E://MavenTest/)下创建工程目录：





将**src文件夹删除**，然后修改pom.xml文件，将**<packaging>jar</packaging>**修改为**<packaging>pom</packaging>**，**pom表示它是一个被继承的模块**，修改后的内容如下：

# 创建system-domain模块

同样利用mvn archetype:generate命令创建工程。

进入system-parent目录：

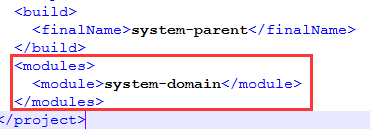
**mvn archetype:generate** -DgroupId=me.gacl -DartifactId=**system-domain** -DarchetypeArtifactId= **maven-archetype-webapp** -DinteractiveMode=false

若失败，利用交互模式。

mvn archetype:generate -DarchetypeArtifactId=maven-archetype-webapp

(注意：system-parent的pom.xml文件必须修改为**<packaging>pom</packaging>**，否则创建不成功)

同时，在system-parent目录中的pom.xml文件自动添加了如下内容：



修改**system-domai**n目录中的pom.xml文件，把**<groupId>me.gacl</groupId>和<version>1.0-SNAPSHOT</version>**去掉，加上<packaging>jar</packaging>，因为groupId和version会继承system-parent中的groupId和version，packaging设置打包方式为jar。



注意：删除的是system-domain自己的groupId和version，不要删除parent中的。

# 创建system-dao模块

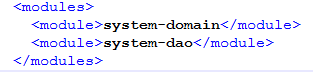
进入system—parent目录：创建system-dao工程：

**mvn archetype:generate** -DgroupId=me.gacl -DartifactId=**system-dao** -DarchetypeArtifactId= **maven-archetype-webapp** -DinteractiveMode=false

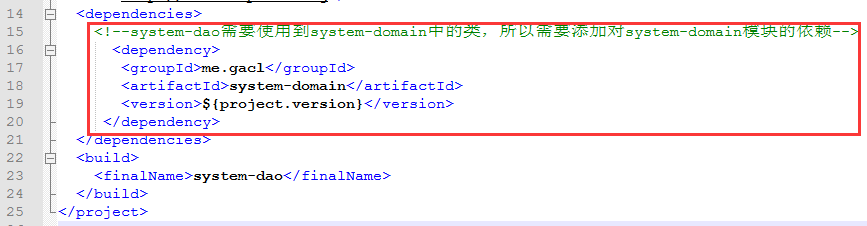
若失败，利用交互模式。

mvn archetype:generate -DarchetypeArtifactId=maven-archetype-webapp

创建成功后，system-parent的pom.xml文件中增加了module：



　修改system-dao目录中的pom.xml文件，同理，把<groupId>me.gacl</groupId>和<version>1.0-SNAPSHOT</version>去掉，加上**<packaging>jar</packaging>**，因为groupId和version会继承system-parent中的groupId和version，**packaging设置打包方式为jar**，**同时添加对system-domain模块的依赖**，修改后的内容如下：



# 创建system-service模块

下面大体过程与创建system-dao类似，不同之处就是添加依赖不同。

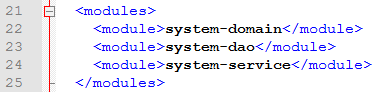
进入system—parent目录：创建system-service工程：

**mvn archetype:generate** -DgroupId=me.gacl -DartifactId=**system-service** -DarchetypeArtifactId= **maven-archetype-webapp** -DinteractiveMode=false

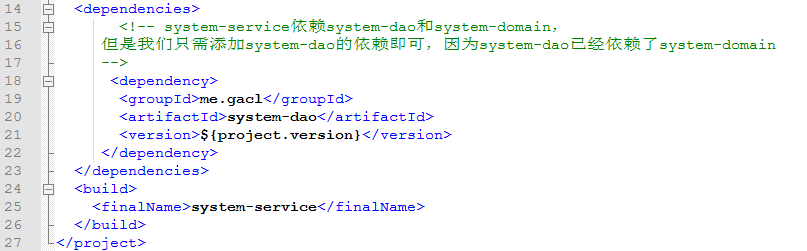
若失败，利用交互模式。

mvn archetype:generate -DarchetypeArtifactId=maven-archetype-webapp

创建成功后，system-parent的pom.xml文件中增加了module：



　修改system-service目录中的pom.xml文件，，把<groupId>me.gacl</groupId>和<version>1.0-SNAPSHOT</version>去掉，加上**<packaging>jar</packaging>，**因为groupId和version会继承system-parent中的groupId和version，**packaging设置打包方式为jar，同时添加对system-dao模块的依赖，system-service依赖system-dao和system-domain，但是我们只需添加system-dao的依赖即可，因为system-dao已经依赖了system-domain。**



# 创建system-web模块

以上工程包括parent都可以建为普通的Maven工程，但是该工程必须为web工程。

-DarchetypeArtifactId=**maven-archetype-webapp**

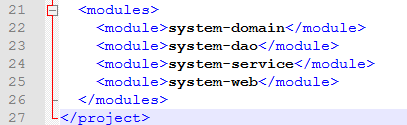
进入system—parent目录：创建system-web工程：

**mvn archetype:generate** -DgroupId=me.gacl -DartifactId=**system-dao** -DarchetypeArtifactId= **maven-archetype-webapp** -DinteractiveMode=false

若失败，利用交互模式。

mvn archetype:generate -DarchetypeArtifactId=maven-archetype-webapp

创建成功后，system-parent的pom.xml文件中增加了module：



　修改system-web目录中的pom.xml文件，，把**<groupId>me.gacl</groupId>**和**<version>1.0-SNAPSHOT</version>**去掉，因为groupId和version会继承system-parent中的groupId和version，**同时添加对system-service模块的依赖**。 注意，**web项目的打包方式是war**，不是jar。



# 编译运行项目

经过上面的五个步骤，相关的模块全部创建完成，怎么运行起来呢。由于最终运行的是system-web模块，所以我们对该模块添加jetty支持，方便测试运行。修改system-web项目的pom.xml如下：

　在命令行进入system-parent目录，然后执行下列命令：

mvn clean install

命令执行完后，在system-web目录下多出了target目录，里面有了system-web.war，如下图所示：

# 导入import到Eclipse中开发

在项目浏览框右击，选中import，选中Existing Maven Projects。

